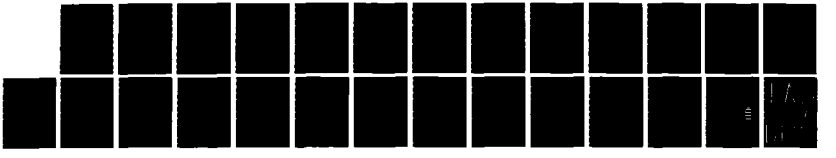
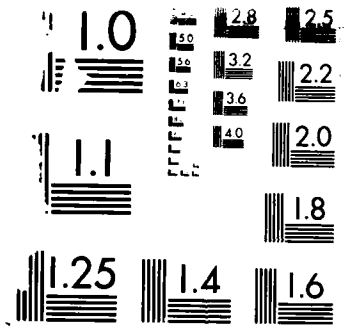
AD-A174 920 A VIRTUAL TERMINAL PROGRAM FOR INTER-COMPUTER SYSTEM COMMUNICATIONS UNDER.. (U) ARMY BALLISTIC RESEARCH LAB ABERDEEN PROVING GROUND MD M A'DEWILDE SEP 86 1/1

UNCLASSIFIED BRL-MR-3546 F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

(12)

AD

MEMORANDUM REPORT BRL-MR-3546

A VIRTUAL TERMINAL PROGRAM FOR INTER-COMPUTER SYSTEM COMMUNICATIONS UNDER THE RT-11 OPERATING SYSTEM

Mark A. DeWilde

September 1986

US ARMY BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U. S. Department of Commerce, Springfield, Virginia 22161.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Memorandum Report BRL-MR-3546	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A VIRTUAL TERMINAL PROGRAM FOR INTER-COMPUTER SYSTEM COMMUNICATIONS UNDER THE RT-11 OPERATING SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Mark A. DeWilde		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Ballistic Research Laboratory ATTN: SLCBR-IB Aberdeen Proving Ground, MD 21005-5066		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1L161102AH43
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Ballistic Research Laboratory ATTN: SLCBR-DD-T Aberdeen Proving Ground, MD 21005-5066		12. REPORT DATE September 1986 13. NUMBER OF PAGES 23
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NA

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19 KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Virtual Terminal
Inter-Computer Communications
PDP-11/RT-11 Computer

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

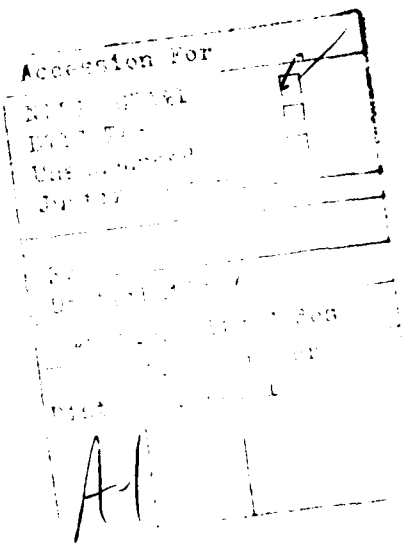
A program is described for performing file transfers between a DEC PDP-11 computer under the RT-11 operating system and another interactive computer system. The program provides for transparent mode operation, uploading and downloading of ASCII files, and data transfer to remote machines. The sole requirement at the host is the availability of a text editor program, and a command to print a file on a user screen.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

TABLE OF CONTENTS



I. INTRODUCTION

The RT-11 operating system is a general purpose single/dual job, single-user operating system for real time tasks on Digital Equipment Corporation's Model PDP-11 computers. Several of these systems were used in the laboratory for automated data acquisition and control applications.[1,2,3,4] Since these computers are small and not particularly well suited for data reduction applications, larger off-site machines were used for this purpose. Data transfer took the form of writing the data onto floppy discs and physically transporting these to the computer site for reading by the time share systems. In most cases, the data sets were too large to be manually typed into the programs, but were small enough to be transferred over communications lines to the remote host machine. The program covered in this report was designed primarily to do this function. As an additional benefit, this program made it possible to do program development on the timeshare systems, and then downloading the programs to the local machine for recompilation and execution. The single-user machines are thus released for their intended tasks rather than being used for program development. Additional benefits were the ability to use the local machine's graphic output devices to display and manipulate output from the analysis programs, and to be able to do word processing tasks locally when the host machines were unavailable.

II. ALGORITHMS

The hardware configuration required for the program is simply two RS-232 communications ports, one for the system console, and one to connect to whatever type of communications are available to the timeshare system (modem, hard-wire, etc.). The addresses for the interface to this second port are assumed to be 176520 to 176526 (octal). The bit assignments and programming details of the port (type DL11) not covered in this report may be found in Reference 5. Briefly, the four registers are a receiver status register, a receiver data buffer, a transmitter status register, and a transmitter data buffer. In the receiving stage, bit seven of the status register is set whenever a character has been received into the data buffer, and has not been read by the program yet. In the transmitter stage, bit seven of the status register is set whenever the data buffer may be loaded with the character to be sent out.

The first part of the program makes the system act as a "dumb" terminal (hence the name "virtual terminal"). In this mode, the algorithm consists of the following steps:

1. Check to see if the user has typed a character on the console device.

2. If a character is available, wait until the transmitter is readv, and send it to the host. If, however, the character is a pre-defined "flag" character, then call the selection menu subroutine.

3. Check to see if the host has sent a character. Output the character to the console terminal.

4. Go to step 1.

In this scheme, any character may be defined as the "flag" character and can be changed at will to reflect the host system's command characters, so as not to interfere with user-host communications. When in this loop, the system will act as if there were nothing between the console terminal and the host until the flag character is typed. The only exception to this is the "break" condition that may be sent by a terminal, and is required by some communications systems to start a connect procedure. Under the virtual terminal program, this condition cannot be sent by pressing the "break" key, but must instead be supplied as one of the selection menu options.

The selection menu is a list of single letter commands and their meanings that perform various functions: sending a break condition, uploading programs from the local machine to the host, downloading from the host to a local file, and the like. Each of the options on this menu calls a subroutine to perform the function, returning to the virtual terminal mode when finished.

For uploading, a file name is solicited from the user, the file opened and read character by character. Each character is sent to the host, and when a character is echoed the next is sent. The echoed character is not necessarily the same since many systems will return a number of space characters when sent a tab character, or will echo a carriage return and line feed when sent a carriage return, and other such substitutions. Random errors are not checked for or corrected. The idea here is that the host's echoes indicate that it has accepted a character and can probably receive another. In this way, the likelihood of overflowing the host's input buffers is low. As the characters are echoed by the host, they are printed on the console. The user can then tell how far the transfer has progressed, can see if the host sends any error messages, can see if the system manager has sent out any broadcast messages to users, and can tell when the transfer is complete. When completed, the program drops back into virtual terminal mode. To handle the uploaded data at the host and put it into a file, the user uses the transparent mode to start a text editor on the host. The text editor is instructed to accept everything typed as input text, after which the user starts the upload. At the end of the upload, back in virtual terminal mode, the user signals the text editor that the text is done, and instructs it to write the file and exit. The file now exists on the host system.

Downloading of files to the local system is a bit more involved. In order to avoid writing a control program on each host to be used, the near-universal XON/XOFF protocol is used to control the transmission of data from the host. The algorithm consists of first soliciting the user for the output file name. Next, a modified type of virtual terminal mode is entered, wherein any characters that the host sends are printed on the console terminal as before, but in addition, if they are not echoes of characters typed by the user, they are also written to the file the user has specified. While actual file writing is going on, transmission from the host is stopped by sending the XOFF character, after which the program continues accepting characters until a timeout waiting for new characters to be received occurs. At this time, the data buffer is written to disc, the XON character sent to continue the transmission of characters, and the downloading continues. When the end of the downloading occurs (the user looks at the echoed text on the console screen to see this), the "flag" character is typed, causing the local disc file to be closed, and the virtual terminal mode to be entered. In operation, the download mode is established, and a command sent to the host to type the

file to the terminal. In this way, no special routine is needed on the host to download, and large data buffers within the program that might overflow on large transfers are unnecessary. The host must, however, support XON/XOFF protocol for this mode to work correctly. The remaining selections on the menu are to return directly to the virtual terminal mode in case the menu was entered by accident, and then exit from program mode. This is necessary since the normal abort characters on the system are disabled in order not to have any characters except the "flag" character intercepted by the program.

III. IMPLEMENTATION

The programs are written in MACRO-11, the assembly language of the PDP-11/RT-11 system. While higher level languages could have been used, this choice insures the fastest response possible on the machine, and for this level of program is not excessively taxing in the writing. To begin, the terminal handler is set to "special mode," in which characters typed are not echoed to the terminal, and are immediately available to the user program. Next the abort character recognition is turned off, and the sign on message typed. The virtual terminal loop described above is entered, and continues until the flag character is typed. One note to be made concerns the terminal handler's response to the typing of a carriage return. The handler not only sends the carriage return to the user program, but also a line feed. The terminal handler on the remote system also accepts a carriage return and echoes both carriage return and line feed. For this reason, when the local handler sends a carriage return and line feed to the program, only the carriage return is sent to the host. Both the return and line feed are sent to the screen when echoed from the host, however. When the menu is typed on the screen, the terminal is still in special mode so that when the selection is typed, the execution begins immediately without waiting for a carriage return to be typed. If lower case letters are typed, they are converted to upper case before being interpreted by the menu selection code. Any unknown selection, or a carriage return will cause the redisplay of the menu.

When the "break" condition transmission is chosen, the bit in the transmitter status register corresponding to this state is set, a timing loop entered to continue sending this condition for a period of time, and then the bit in the status register is cleared. The user is sent a message indicating that the break is being sent. When the upload option is chosen, the first action is to set the terminal back to normal mode, echoing the characters as typed, and waiting for a carriage return before transmitting to the program. The file name for input is then prompted for and the .CSIGEN macro called in order to open the file for input. Next, a block pointer is set to zero to read the first block of the file, the data buffer set up, and the block read. If errors occur, messages are sent to the user, and the virtual terminal mode reentered. If the read went well, then character by character (with the exception of line feeds as before) the text is sent to the host until a null character indicating end of text is encountered. The echoed characters from the host are sent to the terminal. As mentioned previously, at least one character must be received in echo before the next character will be sent. If no echo occurs for a period of 75 seconds (approximately) a time-out waiting for echo error will be printed on the terminal, the upload aborted and a return to the virtual terminal mode made. This is usually sufficient in most cases for even the most heavily loaded systems to respond.

The flow of the download program is a bit more complicated. To begin with, the registers to be used are all preserved in storage locations. Two contiguous 256-word (1 block each) buffers are set up, along with a pointer (R3) to the next open buffer location (byte). The terminal handler is set to normal mode, the output file name prompted for and input, and the file opened for write by the macro call .CSIGEN as before. A note of the peculiarities of RT-11 is due here. The output file name must be followed by an equal sign (=) in order to be used as an output file. Next, a pointer is set to read the first block of the data file, the data buffer set up to be written, and a special variation of the virtual terminal mode entered (at label ENTER1). At this point the user is talking to the host, and gives the command that will dump a file to the screen. The virtual terminal handler in this section echoes all characters to the screen, and also keeps track of the last one typed by the user. If a character is received that was not typed by the user, it is put into the buffer where R3 points. When R3 points to the end of the first 256-word buffer, a routine is called that sends an XOFF character to the host, and then enters a timed wait for character loop. Each time a character is received after sending the XOFF, the timing is reinitialized. When no character is received before timeout, the return to the calling routine is made. These characters received after the XOFF are stored in the other buffer, and echoed to the terminal. The routine that called the XOFF routine again has control, and initiates the writing of the filled buffer to the disc file. The second buffer is filled the same way, but when the XOFF is sent, the pointer R3 is reset to point to the start of the first buffer so that the initial condition is re-established. After the write is accomplished, a call is made to a routine that sends an XON character to the host, and the download virtual terminal special mode is reentered. When the file is done being downloaded, the flag character is typed, and this causes the program to drop into a routine to put a null byte after the last character sent by the host to the buffer, and the last block to be written. Then the file is closed, and a return to the original virtual terminal routine made (after restoring the saved registers).

IV. SOURCE CODE

The actual source code for the utility is listed in this section. The routines were assembled by the MACRO assembler, and linked into a runnable module with the system link program. For definitions of the system macro calls such as .TTINR, .TTYOUT, etc., see Reference 6.

```
.TITLE  VTERM
.GLOBL  VTERM
.MCALL  .REGDEF,.TTINR,.TTYOUT,.EXIT,.PRINT,.SCCA,.TTYIN,.CSIGEN
.MCALL  .READW,.WRITW,.CLOSE,.TTOUTR
.REGDEF
.ENABL  LSB
RCSR = 176520           ;THIS CAN BE CHANGED IF THE PORT ADDRESS
RBUF = RCSR + 2         ;IS NOT 176520-176526 (OCTAL)
XCSR = RCSR + 4
XBUF = RCSR + 6
ERRBYT = 52
FLAG = 140              ;FLAG CHARACTER IS
```

```
VTERM:  .SCCA   #MAREA,#SCCA    ;DISABLE  C SEQUENCES SPECIAL EFFECT
        .PRINT  #MESG           ;SIGN ON...
        BIS     #70000,@#44     ;SET TO SPECIAL MODE
ENTER:  .TTINR                  ;GET A TERMINAL CHARACTER IF HANDY
        BCS     1$              ;GO TO 1$ IF NONE AVAILABLE
        CMP     #FLAG,R0        ;IS IT THE FLAG CHARACTER?
        BEQ     MENU            ;GO TO MENU IF IT IS
        CMP     #15,R0          ;IS IT A CR?
        BNE     2$              ;GO TO 2$ IF IT'S NOT
        .TTINR                  ;GET THE UNDESIRED LF
        MOV     #15,R0          ;PUT BACK IN THE CR
2$:     BIT     #200,@#XCSR     ;READY TO XMIT?
        BEQ     2$              ;LOOP UNTIL IS READY
        MOVB    R0,@#XBUF       ;SEND IT OUT
1$:     BIT     #200,@#RCSR     ;GOT A PORT INPUT CHARACTER?
        BEQ     ENTER           ;NO? GO TO ENTER
        MOVB    @#RBUF,R0       ;GET READY TO SEND TO TERMINAL
        .TTYOUT                 ;SEND IT
        BR      ENTER           ;GO ON....
MENU:   .PRINT  #ITEM1          ;PUT THE MENU ON THE SCREEN...
        .PRINT  #ITEM2
        .PRINT  #ITEM3
        .PRINT  #ITEM4
        .PRINT  #ITEM5
        .PRINT  #ITEM6
        .PRINT  #ITEM7          ;SOLICIT THE SELECTION
        .TTYIN                  ;GET IT
        BIC     #40,R0          ;CONVERT TO UPPER CASE
        MOV     R0,R1           ;STORE IN R1
        .TTINR                  ;GET CR
        .TTINR                  ;GET LF
        MOV     R1,R0           ;RESTORE TO R0
        CMP     #102,R0         ;IS IT A "B"?
        BEQ     BREAK           ;IF SO, GO TO BREAK
        CMP     #105,R0         ;IS IT AN "E"?
        BEQ     DONE            ;GO TO DONE IF IT IS
        CMP     #125,R0         ;IS IT A "U"?
        BEQ     UPLOAD          ;IF SO, GO TO ROUTINE UPLOAD
        CMP     #104,R0         ;IS IT A "D"?
        BEQ     6$              ;GO TO ROUTINE DOWNLOAD IF YES
        CMP     #122,R0         ;IS IT AN "R"?
        BEQ     VTERM           ;GO TO VTERM IF IT IS
        BR      MENU            ;PRINT MENU ON OTHER SELECTIONS
DONE:   .EXIT                   ;RETURN TO RT-11 MONITOR
6$:     JMP     DNLOAD          ;START UP THE DOWNLOAD PROGRAM
BREAK:  .PRINT  #BRKMSG         ;TELL USER SENDING BREAK
        MOV     #177777,R0      ;SET UP TIMER
        BIS     #1,@#XCSR       ;START SENDING BREAK
BRKLP:  MOV     #2,R1           ;DO DOUBLE TIMING LOOP
INLP1:  DEC     R1              ;TO CONTINUE SENDING BREAK
        BNE     INLP1
        DEC     R0
        BNE     BRKLP
        BIC     #1,@#XCSR       ;STOP BREAK CONDITION
```

```
        BR      VTERM           ;GO TO VTERM
BRKMSG: .ASCIZ  /Sending Break Condition../
        .EVEN

UPLOAD: BIC     #70000,@#44     ;BACK TO NORMAL MODE
        .PRINT  #PROMPT         ;ASK THE USER FOR INPUT FILE NAME
        .CSIGEN #DEVSPC,#DEFEXT ;GET THE INPUT FILE NAME
        CLR     IOBLK           ;SET FILE BLOCK POINTER TO 0
        MOV     #AREA,R5        ;SET UP TO READ FILE
READ:   .READW  #AREA,#3        ;READ A BLOCK IN
        BCC     OKREAD          ;GO TO OKREAD IF NO READ ERRORS
        TSTB    @#ERRBYT        ;IF ERRORS, FIND OUT WHAT KIND
        BEQ     EOFERR          ;ERRBYT = 0 IS END OF FILE ERROR
        .PRINT  #RERROR         ;IF OTHER READ ERROR,
        BR      MENU            ;GO TO MENU MODE AGAIN
RERROR: .ASCIZ  / ?? READ ERROR ??/
        .EVEN
EOFERR: .PRINT  #DONMSG         ;IF READING PAST END OF FILE,
        JMP     VTERM           ;GO BACK TO ROUTINE VTERM
DONMSG: .BYTE   12
        .BYTE   15
        .BYTE   7
        .ASCIZ  /DONE READING INPUT FILE/
        .EVEN

EOTEXT: MOV     #TXTDON,R1      ;THIS PRINTS OUT THE NORMAL MESSAGE FOR
NXTCHR: MOVB    (R1)+,R0        ;END OF FILE UPLOAD.
        BEQ     EOTRTN
        .TTYOUT
        BR      NXTCHR
EOTRTN: JMP     VTERM
TXTDON: .BYTE   12
        .BYTE   15
        .BYTE   7
        .ASCIZ  /NULL BYTE DETECTED IN INPUT STREAM/
        .EVEN


OKREAD: INC     IOBLK           ;POINT TO NEXT BLOCK TO BE READ
        MOV     #BUFFER,R1      ;MOVE ADDRESS OF DATA READ TO R1
3$:     BIT     #200,@#XCSR     ;SEE IF READY TO TRANSMIT
        BEQ     3$              ;LOOP UNTIL WE ARE
        CMPB    #0,(R1)         ;SEE IF DATA BYTE IS NULL BYTE
        BEQ     EOTEXT          ;DONE WITH TRANSMISSION IF SO
        MOVB    (R1),R0         ;SEND CHARACTER TO TERMINAL
        .TTYOUT
        CMPB    R0,#12          ;IS IT A LF?
        BNE     11$             ;GO TO 11$ IF NOT
        INC     R1              ;ELSE, SKIP IT
        BR      5$
11$:    MOVB    (R1)+,@#XBUF    ;SEND OUT THE NEXT CHARACTER TO HOST
        JSR     PC,INCHR        ;GET INPUT ECHO
        CMPB    #15,R0          ;WAS TRANSMITTED CHAR A CR?
        BNE     5$              ;GO TO 5$ IF NOT
```

```
        JSR     PC,INCHR        ;GET THE ECHOED LF
5$:     CMP     R1,#NDBUFF      ;SEE IF END OF DATA BUFFER
        BEQ     10$             ;READ IN NEXT BLOCK IF SO
        BR      3$              ;ELSE, SEND SOME MORE BYTES OUT
10$:    JMP     READ

AREA:   .WORD   0               ;DATA STRUCTURES FOR READ REQUEST
IOBLK:  .WORD   0
        .WORD   BUFFER
        .WORD   256.
        .WORD   0
BUFFER: .BLKW   256.
NDBUFF = .

PROMPT: .BYTE   12              ;PROMPT TO SEND TO USER
        .BYTE   15
        .ASCII  /ENTER INPUT FILE NAME >>/
        .BYTE   200
        .EVEN

MESG:   .BYTE   12              ;NORMAL SIGN-ON MESSAGE
        .BYTE   15
        .ASCIZ  /Virtual Terminal program... type " " to exit../
        .EVEN

MAREA:  .BLKW   5               ;DATA STRUCTURE FOR .SCCA CALL
SCCA:   .WORD   0


;       MENU STRINGS.......

ITEM1:  .BYTE   12
        .BYTE   15
        .ASCIZ  /Selection Mode.. enter choice/
        .EVEN
ITEM2:  .ASCIZ  /       B = Send Break Condition to Remote System/
        .EVEN
ITEM3:  .ASCIZ  /       E = Exit From Program Vterm/
        .EVEN
ITEM4:  .ASCII  /       U = Upload Disc File To Remote System ** NOTE **/
        .BYTE   12
        .BYTE   15
        .ASCII  /           remote system must be running editor in insert/
        .BYTE   12
        .BYTE   15
        .ASCIZ  /           text mode before starting this selection./
        .EVEN
ITEM5:  .ASCIZ  /       D = Download File From Remote System to Disc File/
        .EVEN
ITEM6:  .ASCIZ  /       R = Return to Virtual Terminal Mode/
        .EVEN
ITEM7:  .ASCII  /   Selection >>/
        .BYTE   200
        .EVEN
```

```
DEFEXT: .WORD   0               ;DEFAULT EXTENSIONS FOR .CSIGEN CALL
        .WORD   0
        .WORD   0
        .WORD   0

;************** ROUTINE TO GET INPUT CHARACTER ***************

INCHR:  MOV     #20,R4
9$:     MOV     #177777,R5
7$:     DEC     R5
        BEQ     8$
        BIT     #200,@#RCSR     ;CHARACTER IN INPUT BUFFER?
        BEQ     7$              ;LOOP TIL THERE IS
        MOVB    @#RBUF,R2       ;GET THE INPUT CHARACTER
        RTS     PC              ;AND RETURN
8$:     DEC     R4
        BNE     9$
        .PRINT  #TIMERR
        JMP     VTERM
TIMERR: .BYTE   12
        .BYTE   15
        .BYTE   7
        .ASCIZ  /TIME OUT ERROR WAITING FOR ECHO/
        .EVEN

;*************************************************************************
        .DSABL  LSB


        .ENABL  LSB

DNLOAD: MOV     R0,S0           ;PRESERVE REGISTERS
        MOV     R1,S1
        MOV     R2,S2
        MOV     R3,S3
        MOV     #0,IOBLK1       ;START WITH BLOCK 0 OF FILE
        MOV     #0,@#WCNT1      ;SET WORDCOUNT = 0
        MOV     #BUFFR1,IOBUF1  ;SET WRITE BUFFER TO BUFFER 1
        BIC     #70000,@#44     ;BACK TO NORMAL MODE
        MOV     #BUFFR1,R3      ;SET POINTER TO BUFFER 1
        .PRINT  #MESG1          ;SIGN ON...
GETNAM: .PRINT  #PROMP1
        .CSIGEN #DEVSPC,#DEFEX1 ;GET THE INPUT FILE NAME
        CLR     IOBLK1
        BIS     #70000,@#44     ;SET TO SPECIAL MODE
ENTER1: CLR     R1              ;SO IF NO CHARACTER AVAILABLE R1 = NULL
        .TTINR                  ;GET A TERMINAL CHARACTER IF HANDY
        BCS     1$              ;GO TO 1$ IF NONE AVAILABLE
        MOV     R0,R1           ;STORE IT IN R1
        CMP     #FLAG,R1        ;IS IT THE FLAG CHARACTER?
        BNE 2$                  ;GO TO 2$ IF NOT
        JMP     EXITPT          ;GO TO EXIT POINT IF SO
2$:     CMP     #12,R1          ;IS IT A LF?
        BEQ     1$              ;GO TO 1$ IF IT IS. I.E., IGNORE IT
```

```
        BIT     #200,@#XCSR     ;READY TO TRANSMIT?
        BEQ     2$              ;LOOP UNTIL READY
        MOVB    R1,@#XBUF       ;SEND IT OUT
1$:     BIT     #200,@#RCSR     ;GOT A PORT INPUT CHARACTER?
        BEQ     ENTER1          ;NO? GO TO ENTER
        MOVB    @#RBUF,R0       ;GET READY TO SEND TO TERMINAL
        .TTYOUT                 ;SEND IT
        CMP     R0,R1           ;WAS IT THE SAME AS THAT SENT?
        BEQ     ENTER1          ;REPEAT LOOP IF SO
        JSR     PC,PUTCHR       ;ELSE, PUT IT INTO THE DOWNLOAD FILE
        JMP     ENTER1          ;AND REPEAT LOOP
EXITPT: CMP     R3,#NDBUF1      ;SEE IF POINTING INTO BUFFER 2
        BGT     DUMP2           ;AND GO TO DUMP2 IF IT IS
        CMP     R3,BUFFR1       ;SEE IF EMPTY BUFFER
        BNE     WRTBUF          ;GO TO WRTBUF IF NOT EMPTY
        CLR     (R3)            ;END THE BUFFER WITH NULL BYTE
        MOV     #1,WCNT1
        MOV     #BUFFR1,IOBUF1  ;SET UP TO WRITE BUFFER 1
        .WRITW  #AREA1,#0       ;WRITE THE DATA OUT
        BCS     WERR
        .CLOSE  #0              ;CLOSE THE CHANNEL
        BR      RSTOR           ;GO AND RESTORE REGISTERS
WRTBUF: MOVB    #0,(R3)         ;END BUFFER WITH NULL BYTE
        SUB     #BUFFR1,R3      ;R3 IS NOW BYTE COUNT
        INC     R3              ;SINCE WE WANT TO SEND ALL OF 'EM
        ASR     R3              ;NOW IT'S WORD COUNT
        INC     R3              ;JUST IN CASE WE TRUNCATED
        MOV     R3,WCNT1        ;SEND WORD COUNT OUT
        CMP     #256.,WCNT1     ;IS IT GREATER THAN 256?
        BLE     OK              ;GO TO OK IF SO
        DEC     WCNT1           ;DECREMENT IF NOT
OK:     MOV     #BUFFR1,IOBUF1  ;SET UP TO WRITE BUFFER 1
        .WRITW  #AREA1,#0       ;WRITE THE DATA OUT
        BCS     WERR
        .CLOSE  #0              ;CLOSE THE CHANNEL
        BR      RSTOR           ;GO AND RESTORE REGISTERS
DUMP2:  MOVB    #0,(R3)         ;END BUFFER WITH NULL BYTE
        SUB     #NDBUF1,R3      ;R3 IS NOW BYTE COUNT
        ASR     R3              ;NOW IT'S WORD COUNT
        INC     R3              ;JUST IN CASE WE TRUNCATED
        MOV     R3,WCNT1        ;SEND WORD COUNT OUT
        MOV     #BUFFR2,IOBUF1  ;SET UP TO WRITE BUFFER 1
        .WRITW  #AREA1,#0       ;WRITE THE DATA OUT
        INC     @#IOBLK1        ;BUMP THE BLOCK COUNTER
        BCS     WERR
        .CLOSE  #0              ;CLOSE THE CHANNEL
RSTOR:  MOV     S0,R0           ;RESTORE REGISTERS
        MOV     S1,R1
        MOV     S2,R2
        MOV     S3,R3
        JMP     VTERM
WERR:   .PRINT  #WRTERR
        RTS     PC
;******************** SUBROUTINE PUTCHR ************************
```

```
PUTCHR: MOVB    R0,(R3)+        ;PUT THE INPUT CHARACTER IN THE BUFFER
        CMP     R3,#NDBUF1      ;END OF BUFFER 1?
        BNE     3$              ;GO TO 3$ IF NOT
        JSR     PC,XOFF         ;GO TO XOFF IF IT IS
        MOV     #BUFFR1,IOBUF1  ;SET UP TO WRITE BUFFER 1
        MOV     #256.,WCNT1     ;SET WORD COUNT = 256
        .WRITW  #AREA1,#0       ;WRITE THE BLOCK TO DISC
        BCC     8$
        .PRINT  #WRTERR
8$:     INC     IOBLK1          ;POINT TO NEXT BLOCK TO BE READ
        JSR     PC,XON          ;SEND XON
DONPUT: RTS     PC              ;RETURN TO CALLER
3$:     CMP     R3,#NDBUF2      ;END OF BUFFER 2?
        BNE     DONPUT          ;GO TO DONPUTSTET
        MOV     #BUFFR1,R3      ;SET UP POINTER TO BUFFER 1
        JSR     PC,XOFF         ;STOP TRANSMISSION
        MOV     #BUFFR2,IOBUF1  ;SET UP TO WRITE BUFFER 2
        MOV     #256.,WCNT1     ;SET WORD COUNT = 256
        .WRITW  #AREA1,#0       ;WRITE THE BUFFER TO DISC
        BCC     4$
        .PRINT  #WRTERR
4$:     INC     IOBLK1          ;POINT TO NEXT BLOCK TO BE READ
        JSR     PC,XON          ;SEND XON
        BR      DONPUT          ;GO TO DONE PUT CHARACTER

;************** SUBROUTINE XOFF *****************************

XOFF:   BIT     #200,@#XCSR     ;READY TO SEND?
        BEQ     XOFF            ;LOOP TIL WE ARE
        MOVB    #23,@#XBUF      ;SEND XOFF (DC3)
7$:     MOV     #177777,R2      ;SET UP TIMER
5$:     DEC     R2              ;BEGIN CATCHING ANY POST-XOFF STUFF
        BEQ     6$              ;GO TO 6$ IF TIMEOUT
        BIT     #200,@#RCSR     ;CHARACTER AVAILABLE?
        BEQ     5$
        MOVB    @#RBUF,R0       ;PUT THE CHARACTER IN R0
        .TTOUTR                 ;SEND TO TERMINAL
        MOVB    R0,(R3)+        ;GET THE CHARACTER INTO THE BUFFER
        BR      7$              ;GO TO 7$, AND WAIT FOR ANOTHER
6$:     RTS     PC              ;RETURN TO CALLER

;************** SUBROUTINE XON ******************************

XON:    BIT     #200,@#XCSR     ;READY TO XMIT?
        BEQ     XON             ;LOOP TIL READY
        MOVB    #21,@#XBUF      ;SEND XON (DC1)
        RTS     PC

;************** TABLES AND STUFF ****************************


AREA1:  .WORD   0
IOBLK1: .WORD   0
```

```
IOBUF1: .WORD   0
WCNT1:  .WORD   0
        .WORD   0
BUFFR1: .BLKW   256.
NDBUF1 = .
BUFFR2: .BLKW   256.
NDBUF2 = .
DEFEX1: .WORD   0
        .WORD   0
        .WORD   0
        .WORD   0
WRTERR: .BYTE   12
        .BYTE   15
        .ASCIZ  / ? WRITE ERROR ? /
        .EVEN
PROMP1: .BYTE   12
        .BYTE   15
        .ASCII  /TYPE IN THE OUTPUT FILE NAME >>/
        .BYTE   200
        .EVEN
S0:     .WORD   0
S1:     .WORD   0
S2:     .WORD   0
S3:     .WORD   0
MESG1:  .BYTE   12
        .BYTE   15
        .ASCII  /DOWNLOAD FILE FROM REMOTE SYSTEM ROUTINE/
        .BYTE   12
        .BYTE   15
        .ASCII  /ENTER OUTPUT FILE NAME, THEN TYPE THE COMMAND TO THE/
        .BYTE   12
        .BYTE   15
        .ASCII  /REMOTE COMPUTER TO TYPE FILE TO SCREEN. WHEN DONE, TYPE/
        .BYTE   12
        .BYTE   15
        .ASCII  /" " TO CLOSE FILE AND RETURN TO VIRTUAL TERMINAL MODE/
        .BYTE   12
        .BYTE   15
        .BYTE   0
        .EVEN
        .DSABL  LSB
DEVSPC: .WORD   0       ;SPACE FOR DEVICE HANDLERS
        .END
```

V. CLOSING COMMENTS

This utility provides an easy way to establish machine to machine communications in a user friendly way. If the user programs on the systems store their data in the form of ASCII characters in files, this routine can be used to upload/download data files of moderate size quite easily. This utility was created to augment the systems described in Reference 1, and fill one of the gaps (time share system to real time system file transfer) that was indicated by Reference 4.

REFERENCES

1. DeWilde, M.A. and Weaver, C.E., "The Combustion Diagnostics Laboratory Data Acquisition and Control System," BRL Report in press, 1986.

2. DeWilde, M.A., "An Automated System for the Control of and Data Acquisition from Multiphoton Ionization and Florescence Lifetime Measurements," BRL Report in press, 1986.

3. DeWilde, M.A. and Weaver, C.E., "A Versatile Automated Data Acquisition and Control System for Raman and Other Forms of Optical Spectroscopy," BRL-TR in progress, 1986.

4. DeWilde, M.A., "The Human Engineering of Laboratory Data Acquisition and Control Systems," ARBRL-TR-2623, December 1984.

5. Microcomputer Interfaces Handbook, The Digital Equipment Corp., Publication No. EB-20175-20/81, Maynard, MA, 1980.

6. RT-11 Version 4 Documentation Kit, The Digital Equipment Corp., Publication No. QJ013-GZ, Maynard, MA, 1982.

DISTRIBUTION LIST

No of Copies	Organization
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22314
1	HQ DA DAMA-ART-M Washington, DC 20310
1	Commander US Army Materiel Cmd. ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander Armament R&D Center USA AMCCOM ATTN: SMCAR-TDC Dover, NJ 07801-5001
2	Commander Armament R&D Center USA AMCCOM ATTN: SMCAR-TSS Dover, NJ 07801-5001
1	Commander US Army Armament, Munitions and Chemical Cmd ATTN: SMCAR-ESP-L Rock Island, IL 61299
1	Director Benet Weapon Laboratory Armament R&D Center USA AMCCOM ATTN: SMCAR-LCB-TL Watervliet, NY 12189
1	Commander USA Aviation Rsch and Development Cmd ATTN: AMSAV-E 4300 Goodfellow Blvd. St. Louis, MO 63120

No. of Copies	Organization
1	Director USA Air Mobility Rsch and Development Lab. Ames Research Center Moffett Field, CA 94035
4	Commander USA Research Officer ATTN: R. Ghirardelli D. Mann R. Singleton R. Shaw Research Triangle Park NC 27709
1	Commander U.S. Army Communications - Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703
1	Commander U.S. Army Communications - Electronics Command (CECOM) CECOM R&D Technical Library ATTN: AMSEL-IM-L B 2700 Fort Monmouth, NJ 07703-5000
2	Commander USA AMCCOM ATTN: DRSMC-LCA-G D.S. Downs J.A. Lannon Dover, NJ 07801
1	Commander USA AMCCOM ATTN: DRSMC-LC, L. Harris Dover, NJ 07801
1	Commander USA AMCCOM ATTN: DRSMC-SCA-T L. Stiefel Dover, NJ 07801

DISTRIBUTION LIST

No of Copies	Organization
1	Commander USA Missile Command Research, Development, Engineering Center ATTN: AMSMI-RD Redstone Arsenal, AL 35898
1	Commander USA Missile & Space Command Intelligence Center ATTN: AIAMS-YDL Redstone Arsenal, AL 35898
2	Commander USA Missile Command ATTN: DRSMI-RK, D.J. Ifshin Redstone Arsenal, AL 35898
1	Commander USA Tank Automotive Cmd ATTN: AMSTA-TSL Warren, MI 48397-5000
1	Director USA TRADOC Systems Analysis ATTN: ATAA-SL WSMR, NM 88002
2	Commandant USA Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905
1	Commander USA Development and Employment Agency ATTN: MODE-TED-SAB Fort Lewis, WA 98433
1	Office of Naval Rsch Department of the Navy ATTN: R.S. Miller, Code 432 800 N. Quincy Street Arlington, VA 22217

No. of Copies	Organization
1	Navy Strategic Systems Project Office ATTN: R.D. Kinert, SP 2721 Washington, DC 20376
1	Commander Naval Air Systems Cmd ATTN: J. Ramnarace, AIR-54111C Washington, DC 20360
3	Commander Naval Ordnance Station ATTN: C. Irish S. Mitchell P.L. Stang, Code 515 Indian Head, MD 20640
1	Commander Naval Surface Weapons Center ATTN: J.L. East, Jr., G-20 Dahlgren, VA 22448
2	Commander Naval Surface Weapons Center ATTN: R. Bernecker, R-13 G.B. Wilmot, R-16 Silver Spring, MD 20910
4	Commander Naval Weapons Center ATTN: R.L. Derr, Code 389 China Lake, CA 93555
2	Commander Naval Weapons Center ATTN: Code 3891, T. Boggs K.J. Graham China Lake, CA 93555
5	Commander Naval Research Lab, Code 6110 ATTN: L. Harvey J. McDonald E. Oran J. Shnur R.J. Doyle, Washington, DC 20375

DISTRIBUTION LIST

No of Copies	Organization
1	Commanding Officer Naval Underwater System Center Weapons Dept. ATTN: R.S. Lazar/Code 36301 Newport, RI 02840
1	Superintendent Naval Postgraduate School Dept. of Aeronautics ATTN: D.W. Netzer Monterey, CA 93940
6	AFRPL (DRSC) ATTN: R. Geisler D. George B. Goshgarian J. Levine W. Roe D. Weaver Edwards AFB, CA 93523
1	Air Force Armament Laboratory ATTN: AFATL/DLODL ATTN: O.K. Heiney Eglin AFB, FL 32542-5000
2	AFOSR ATTN: L.H. Caveny J.M. Tishkoff Bolling Air Force Base Washington, DC 20332
1	AFWL/SUL Kirtland AFB, NM 87117
1	NASA Langley Research Center ATTN: G.B. Northam/MS 168 Hampton, VA 23365
4	National Bureau of Standards ATTN: J. Hastie M. Jacox T. Kashiwagi H. Semerjian US Dept. of Commerce Washington, DC 20234
1	Aerojet Solid Propulsion Company ATTN: P. Micheli Sacramento, CA 95813
1	Applied Combustion Technology, Inc. ATTN: A.M. Varney P.O. Box 17885 Orlando, FL 32860
2	Atlantic Research Corp. ATTN: M.K. King 5390 Cherokee Avenue Alexandria, VA 22314
1	Atlantic Research Corp. ATTN: R.H.W. Waesche 7511 Wellington Road Gainesville, VA 22065
1	AVCO Everett Research laboratory Division ATTN: D. Stickler 2385 Revere Beach Parkway Everett, MA 02149
1	Battelle Memorial Institute Tactical Technology Center ATTN: J. Huggins 505 King Avenue Columbus, OH 43201
2	Exxon Research & Engineering Company ATTN: A. Dean M. Chou P.O. Box 45 Linden, NJ 07036
1	Ford Aerospace and Communications Corp. DIVAD Division Div. Hq., Irvine ATTN: D. Williams Main Street & Ford Road Newport Beach, CA 92663

DISTRIBUTION LIST

No of Copies	Organization
1	General Electric Armament & Electrical Systems ATTN: M.J. Bulman Lakeside Avenue Burlington, VT 05402
1	General Electric Co. ATTN: M. Lapp Schenectady, NY 12301
1	General Electric Ordnance Systems ATTN: J. Mandzy 100 Plastics Avenue Pittsfield, MA 01203
1	General Motors Rsch Lab Physics Department ATTN: R. Teets Warren, MI 48090
3	Hercules, Inc. Alleghany Ballistics Lab. ATTN: R.R. Miller P.O. Box 210 Cumberland, MD 21501
3	Hercules, Inc. Bacchus Works ATTN: K.P. McCarty P.O. Box 98 Magna, UT 84044
1	Hercules, Inc. AFATL/DLDL ATTN: R.L. Simmons Eglin AFB, FL 32542
1	Honeywell, Inc. Defense Systems Div. ATTN: D.E. Broden/ MS MN50-2000 600 2nd Street NE Hopkins, MN 55343
1	IBM Corporation ATTN: A.C. Tam Research Division 5600 Cottle Road San Jose, CA 95193
1	Director Lawrence Livermore National Laboratory ATTN: C. Westbrook Livermore, CA 94550
1	Lockheed Missiles & Space Company ATTN: George Lo 3251 Hanover Street Dept. 52-35/B204/2 Palo Alto, CA 94304
1	Los Alamos National Lab ATTN: B. Nichols T7, MS-B284 P.O. Box 1663 Los Alamos, NM 87545
1	Olin Corporation Smokeless Powder Operations ATTN: R.L. Cook P.O. Box 222 St. Marks, FL 32355
1	Paul Gough Associates ATTN: P.S. Gough 1048 South Street Portsmouth, NH 03801
2	Princeton Combustion Research Labs, Inc. ATTN: M. Summerfield N.A. Messina 475 US Highway One Monmouth Junction, NJ 08852
1	Hughes Aircraft Company ATTN: T.E. Ward 8433 Fallbrook Avenue Canoga Park, CA 91303

DISTRIBUTION LIST

No of Copies	Organization
1	Rockwell International Corporation Rocketdyne Division ATTN: J.E. Flanagan/HB02 6633 Canoga Avenue Canoga Park, CA 91304
3	Sandia National Labs. Combustion Science Dept ATTN: R. Cattolica D. Stephenson P. Mattern Livermore, CA 94550
1	Sandia National Labs. ATTN: M. Smooke Division 8353 Livermore, CA 94550
1	Science Applications, Inc. ATTN: R.B. Edelman 23146 Cumorah Crest Woodland Hills, CA 91364
2	Univ. of California, Santa Barbara Quantum Institute ATTN: K. Schofield M. Steinberg Santa Barbara, CA 93106
1	University of Southern California Dept. of Chemistry ATTN: S. Benson Los Angeles, CA 90007
1	Case Western Reserve University Division of Aerospace Science ATTN: J. Tien Cleveland, OH 44135
1	Cornell University Department of Chemistry ATTN: E. Grant Baker Laboratory Ithaca, NY 14853
1	University of Dayton Research Institute ATTN: D. Campbell AFRPL/PAP Stop 24 Edwards AFB, CA 93523
1	University of Florida Dept. of Chemistry ATTN: J. Winefordner Gainesville, FL 32611
3	Georgia Inst. of Technology School of Aerospace Eng. ATTN: E. Price Atlanta, GA 30332
2	Georgia Institute of Technology School of Aerospace Eng. ATTN: W.C. Strahle B.T. Zinn Atlanta, GA 30332
1	University of Illinois Dept. of Mech. Eng. ATTN: H. Krier 144 MEB, 1206 W.Green Street Urbana, IL 61801
1	Johns Hopkins Univ./APL Chemical Propulsion Information Agency ATTN: T.W. Christian Johns Hopkins Road Laurel, MD 20707
1	University of Minnesota Dept. of Mechanical Eng. ATTN: E. Fletcher Minneapolis, MN 55455
4	Pennsylvania State University Applied Research Lab. ATTN: G.M. Faeth K.k. Kuo H. Palmer M. Micci University Park, PA 16802

DISTRIBUTION LIST

No of Copies	Organization
1	Polytechnic Institute of NY ATTN: S. Lederman Route 110 Farmingdale, NY 11735
2	Princeton University Forrestal Campus Library ATTN: K. Brezinsky I. Glassman P.O. Box 710 Princeton, NJ 08540
1	Princeton University MAE Dept. ATTN: F.A. Williams Princeton, NJ 08544
1	Science Applications, Inc. ATTN: H.S. Pergament 1100 State Road, Bldg. N Princeton, NJ 08540
1	Space Sciences, Inc. ATTN: M. Farber Monrovia, CA 91016
4	SRI International ATTN: S. Barker D. Crosley D. Golden Tech. Lib 333 Ravenwood Avenue Menlo Park, CA 94025
1	Stevens Institute of Technology Davidson Laboratory ATTN: R. McAlevy, III Hoboken, NJ 07030
1	Teledyne McCormack-Selph ATTN: C. Leveritt 3601 Union Road Hollister, CA 95023
10	Central Intelligence Agency Office of Central Reference Dissemination Branch Room GE-47 HQS Washington, D.C. 20502
1	Thiokol Corporation Elkton Division ATTN: W.N. Brundige P.O. Box 241 Elkton, MD 21921
3	Thiokol Corporation Huntsville Division ATTN: D.A. Flanagan Huntsville, AL 35807
3	Thiokol Corporation Wasatch Division ATTN: J.A. Peterson P.O. Box 524 Brigham City, UT 84302
1	United Technologies ATTN: A.C. Eckbreth East Hartford, CT 06108
2	United Technologies Corp. ATTN: R.S. Brown R.O. McLaren P.O. Box 358 Sunnyvale, CA 94088
1	Universal Propulsion Company ATTN: H.J. McSpadden Black Canyon Stage 1 Box 1140 Phoenix, AX 85029
1	Veritay Technology, Inc. ATTN: E.B. Fisher P.O. Box 22 Bowmansville, NY 14026
1	Brigham Young Univ. Dept. of Chemical Eng. ATTN: M.W. Beckstead Provo, UT 84601
1	California Institue of Technology Jet Propulsion Lab. ATTN: MS 125/159 4800 Oak Grove Drive Pasadena, CA 91103

DISTRIBUTION LIST

No of Copies	Organization
1	California Institue of Technology ATTN: F.E.C. Culick/ MC 301-46 204 Karman Lab. Pasadena, CA 91125
1	Univ. of California, Berkeley Mechanical Engineering Department ATTN: J. Daily Berkeley, CA 94720
1	Univ. of California Los Alamos National Laboratory ATTN: T.D. Butler P.O. Box 1663, Mail Stop B216 Los Alamos, NM 87545
2	Purdue University School of Aeronautics and Astronautics ATTN: R. Glick J.R. Osborn Grissom Hall West Lafayette, IN 47907
3	Purdue University School of Mechanical Engineering ATTN: N.M. Laurendeau S.N.B. Murthy D. Sweeney TSPC Chaffee Hall West Lafayette, IN 47906
1	Rensselaer Polytechnic Institute Dept. of Chemical Engineering ATTN: A. Fontijn Troy, NY 12181
2	Southwest Research Institute ATTN: R.E. White A.B. Wenzel 8500 Culebra Road San Antonio, TX 78228
1	Stanford University Dept. of Mechanical Engineering ATTN: R. Hanson Stanford, CA 93106
1	University of Texas Dept. of Chemistry ATTN: W. Gardiner Austin, TX 78712
1	University of Utah Dept. of Chemical Engineering ATTN: G. Flandro Salt Lake City, UT 84112
1	Virginia Polytechnic Institute & State University ATTN: J.A. Schetz Blackburg, VA 24061
	Aberdeen Proving Ground
	Dir, USAMSAA ATTN: AMXSY-D AMXSY-MP, H. Cohen Cdr, USATECOM ATTN: AMSTE-TO-F Cdr, CRDC, AMCCOM ATTN: SMCCR-RSP-A SMCCR-MU SMCCR-SPS-IL

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. BRL Report Number _____ Date of Report _____

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

_____

_____

4. How specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

_____

_____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided or efficiencies achieved, etc? If so, please elaborate. _____

_____

_____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

_____

_____

_____

CURRENT ADDRESS

_____
Name

_____
Organization

_____
Address

_____
City, State, Zip

7. If indicating a Change of Address or Address Correction, please provide the New or Correct Address in Block 6 above and the Old or Incorrect address below.

OLD ADDRESS

_____
Name

_____
Organization

_____
Address

_____
City, State, Zip

(Remove this sheet along the perforation, fold as indicated, staple or tape closed, and mail.)

FOLD HERE

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-5066

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC

POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-9989

FOLD HERE

END

1-87

DTIC